S698MIL-DKit(S698-MIL 应用开发系统)

使用说明书

珠海欧比特控制工程股份有限公司

广东省珠海市港湾大道白沙路 1 号(邮编：519080)

电话：0756-3391979　　传真：0756-3391980

http://www.myorbita.net

前　　言

感谢您使用最新推出的 S698MIL-DKit(S698-MIL 应用开发系统)！

为了使您能尽快熟练地操作 S698MIL-DKit,我们随机配备了内容详细的使用说明书，在您第一次使用时，请务必仔细阅读所有随机资料。

基于提高部件及系统性能和可靠性的需要，我们有时会对系统（包括硬件和软件）做一些改动，届时，我们会尽量修改或增加资料，但仍可能在某些描述上与实际改动后的不一致，敬请谅解。本使用说明书中如有错误和疏漏之处，热切欢迎您的指正。

厂家相关信息

- 生产企业名称:　**珠海欧比特控制工程股份有限公司**
- 生产企业地址:　**广东省珠海市港湾大道白沙路 1 号**
- 产品标准编号:　S698MIL-DKit
- 售后服务单位:　**珠海欧比特控制工程股份有限公司**
- 联系方式:

地　　址:　**广东省珠海市港湾大道白沙路 1 号欧比特科技园**
电　　话:　0756-3391979
传　　真:　0756-3391980
邮　　编:　519080

使用注意事项

- 本系统无客户可自行维修的组件，发生故障时请勿自行拆卸。
- S698MIL- DKit 在出厂前已经过充分的检验和校验。用户在使用前，请先确认它在运输过程中没有受到损坏。
- 使用前请先核对一下您手中的 S698MIL- DKit 与订货时的型号是否一致，配件是否完整。
- 为避免系统受到损坏，请使用系统包装箱进行运输，直至到达使用现场。
- 拨插设备时务必请先断电后再操作。
- 存放地点应具备以下条件：防雨、防潮；机械振动要小，防止可能的碰撞；温度：0℃ ～ 40℃；湿度：40% ～ 80%。

安全预防

- ■ S698MIL-DKit(S698-MIL 应用开发系统)内部的电子部件可能会被静电损坏，为保证系统的安全，当接触这些部件时，请先确保人体没有静电。
- ■ 为了保证操作人员和设备的安全，请仔细阅读该说明书并严格按照安全规则操作。对于用户违反操作规则而造成的一切损失和用户擅自拆装而造成的系统损坏，本公司将不承担责任。
- ■ 若系统出现故障，请及时通知我们，并请提供产品的完整型号、出厂编号、故障现象、使用环境等详细资料，以便我们迅速为您排除故障。

声明

珠海欧比特控制工程股份有限公司拥有此非公开出版的使用说明书的版权，并有权将其作为保密资料处理。本使用说明书只作为操作、保养和维修产品的参考资料，其他人无权向他人公开此使用说明书。

本使用说明书包含由版权法保护的专有资料，版权所有，未经珠海欧比特控制工程股份有限公司的书面同意不得将本使用说明书的任何部分进行照相复制，复印或翻译成其它语言。

本使用说明书包含的内容可以不予通知而有所变更。

制造商的责任

只有在下列情况下，珠海欧比特控制工程股份有限公司才认为应对系统的安全、可靠性和性能的有关问题负责：

- 装配、扩充、重新调整、改进或维修均由公司认可的人员进行操作；
- 系统使用地点的电气安全符合国家标准；
- 系统的使用按操作要求进行。

目　　录

第 1 章　S698MIL-DKIT介绍 1

1.1　概述 1

1.1.1 S698MIL-DKIT简介 1

1.2　功能特点 2

1.2.1 适用范围 3

1.2.2 显示及键盘 3

1.3　基本技术指标 3

1.4　系统框图 3

1.5　布局说明 4

第 2 章　接线与功能模块说明 6

2.1　安装步骤 6

2.2　外部接线 7

2.2.1 DSU、RS232 的接口说明 8

2.2.2 RS422 接口说明 8

2.2.3 RS485 接口说明 8

2.2.4 A/D接口说明 9

2.2.5 D/A接口说明 9

2.3　功能模块说明 9

2.3.1 S698-MIL处理器芯片 9

2.3.2 ROM 10

2.3.3 SRAM 10

2.3.4 NAND FLASH 10

2.3.5 IC卡(又称智能卡) 11

2.3.6 DSU串口 12

2.3.7 LCD显示屏 12

2.3.8 键盘 13

2.3.9 数码管显示 13

2.3.10 A/D和D/A 14

2.3.11 LED..15
2.3.12 CPLD..15
第 3 章 软件系统..16
3.1 交叉开发调试方式..16
3.2 安装Orion 4.0 集成开发环境..16
3.3 Orion 4.0 使用..17
3.3.1 Orion 4.0 开发流程..18
3.4 Orion 4.0 使用说明..19
3.4.1 主窗口..19
3.4.2 选择工作间..20
3.4.3 工程的新建..20
3.4.4 编译调试环境的配置..24
3.4.5 调试运行环境的设置..27
3.4.6 工程的编译、链接..31
3.4.7 程序的调试..31
3.5 DSU调试步骤..36
3.5.1 FLASH程序固化..37
3.6 菜单界面..39
3.6.1 LED界面..39
3.6.2 A/D D/A界面..40
3.6.3 IC卡界面..41
3.6.4 帮助界面..41
第 4 章 维护和故障处理..42
4.1 日常维护..42
4.2 故障处理..42
附录A：订货信息..43
附录B：装箱单..43

第1章 S698MIL-DKIT介绍

1.1 概述

1.1.1S698MIL-DKIT简介

图 1-1 系统外观实物图

S698MIL-DKit 是专业支持 S698-MIL 处理器，且集学习、实验与开发于一体的三合一应用开发系统。选用基于高可靠高性能的 32 位 S698-MIL 处理器作为主 CPU，随机免费配套提供了 Oiron 4.0 集成开发环境及全部功能模块的开发源码。

通过电话、BBS、E-mail 等多种方式为初学者提供免费的技术支持。

通过帮助用户创造价值，依此来赢得用户、社会的支持和信任。

1.2 功能特点

S698MIL-DKIT 系统功能强大、操作简单、界面友好，下面是 S698MIL-DKIT 系统的主要功能特点：

- S698-MIL 处理器
- 240*320 点阵 LCD
- IC 卡读卡器
- 12 位 A/D
- 12 位 D/A
- RS232
- RS422
- RS485
- 8 位 8 段数码管
- 5 个 LED
- 4*4 矩阵键盘
- 16MB NandFlash
- 1MB SRAM
- 512KB FLASH
- CPLD
- DSU(无需仿真器，可支持硬件直接调试口)。

➢ 提供丰富的软件代码

  - 相关功能模块的软件代码；
  - RTMES 操作系统；
  - uCLinux 嵌入式操作系统
  - Orion 4.0 集成开发环境
  - 详细的开发手册

1.2.1适用范围

- S698MIL-DKit 是专业支持 S698-MIL 处理器，且集学习、实验与开发于一体的三合一应用开发系统。是科研开发、教学实验、电子设计创新实验室、现代嵌入式系统实验室的上佳选择。

1.2.2显示及键盘

- 320×240 大屏幕液晶显示；
- 4×4 矩阵键盘方便输入。

1.3 基本技术指标

技术指标

- 供电电压：外接交流电压输入，范围 180VAC~240VAC；
- A/D 电压输入：0～2.5V；
- D/A 电压输出：0～2.5V
- 通讯接口：
  - ✧ 1 路 RS232 总线接口，波特率:1200～115200bps 可选；
  - ✧ 1 路 RS422 总线接口，波特率:1200～115200bps 可选；
  - ✧ 1 路 RS485 总线接口，波特率:1200～115200bps 可选；
  - ✧ 1 路 IC 卡接口；
- 工作温度：-20～+75℃；
- 相对湿度：10%～80%；
- 外形尺寸：420*330*100(mm)；

1.4 系统框图

S698MIL-DKit 系统由 S698-MIL 芯片、丰富的外围接口器件、电源部分、IDE 开发环境以下丰富的开发资料组成。图 1-2 是 S698MIL-DKit 的功能框图，以便用户有个直观的了解整个系统的功能。

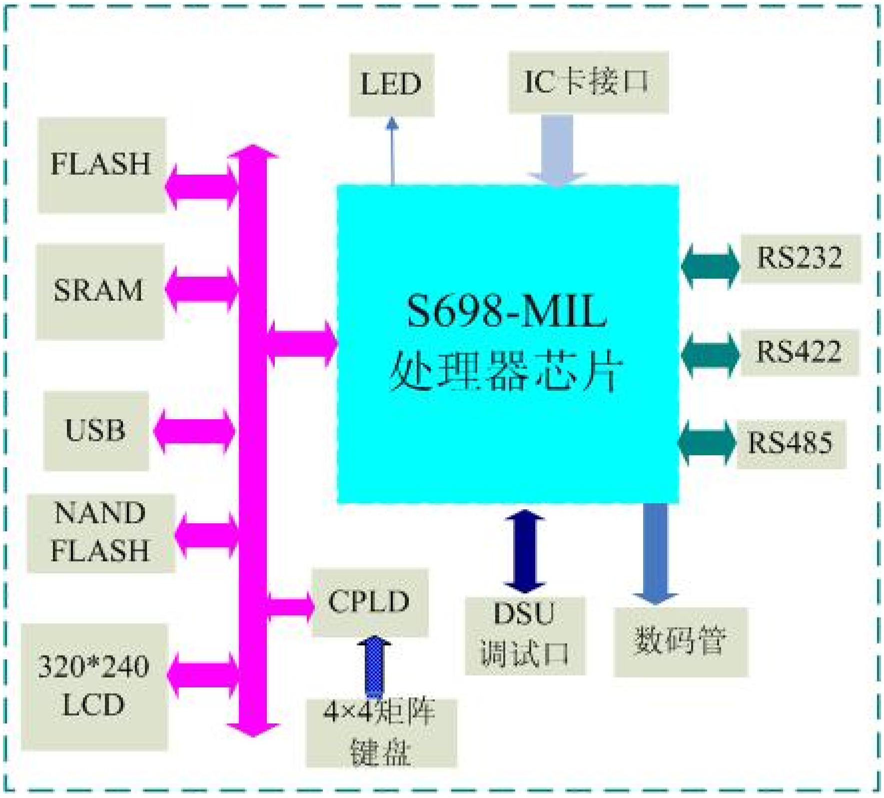


图 1-2 功能框图

1.5 布局说明

1. 在用户了解了功能框图后，图 1-3 详细介绍了本系统的各个功能布局，以方便用户在使用中能够快速找到。

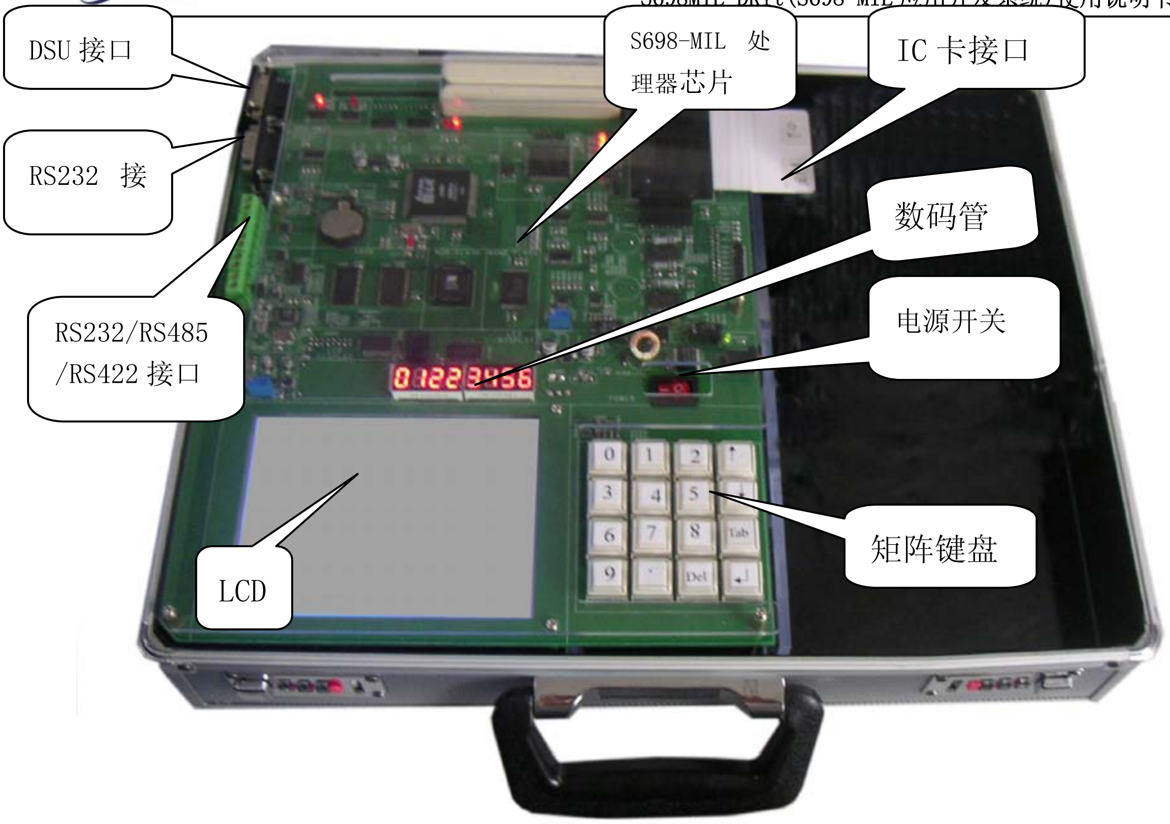

DSU 接口
S698-MIL 处理器芯片
IC 卡接口
RS232 接
数码管
RS232/RS485 /RS422 接口
电源开关
矩阵键盘
LCD

图 1-3 布局图

第2章 接线与功能模块说明

本系统拨插设备时务必请先断电后再操作，使用时请注意用电安全。

2.1 安装步骤

整个系统的安装可按以下步骤进行：

第 1 步：检查产品包装箱；

第 2 步：安装 Orion 4.0 开发环境；

第 3 步：接线；

第 4 步：接通各部分电源；

第 5 步：进行系统操作。

下面具体介绍每一步的操作。

*第 1 步：*检查产品包装箱

在您打开产品的包装箱后应做以下检查：

- 检查包装箱内的设备及配件是否和装箱单上所列的产品型号、数量一致；
- 检查设备是否完好，有没有在运输过程中损坏。如果发生了损坏请立即与运输公司或经销商联系；
- 检查您所收到的设备与您定货单填写的型号、数量是否一致，如有差异请立即与经销商或本公司联系。

*第 2 步：*安装 Orion 4.0 开发环境

详细步骤请参见第 3.2 章。

*第 3 步：*接线

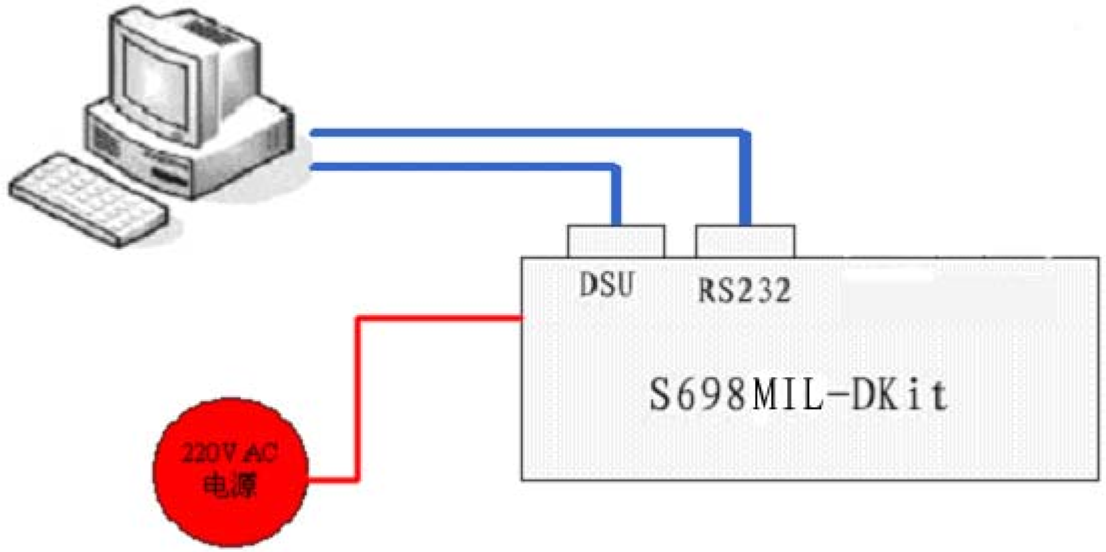


图 2-1 接线方式示意图

本系统供电方式为 220V 交流电源，DSU 调试口、RS232 通过串口线与 PC 机相连。图 2-1 提供了部分功能的接线示意图，使用时除了电源线是必须接的，当不使用其它功能时，可不用接其它连线。

*第 4 步：*接通各部分电源

接线完毕后，认真检查接线是否正确，确认无误后，接通 S698MIL-DKit 的电源。注意电源制式一定要与设备要求相符。

第 5 步： 进行系统操作

通电后，就可以进行系统的设置了。具体的系统操作方法请阅读第 3.4 章。

2.2 外部接线

所有的外部接线(除了电源线)都从系统的左边引出。按照从上往下的顺序，其中包括：

- 一个 DSU 调试接口；
- 一个 RS232 串行接口；
- 一个 RS422 串行接口
- 一个 RS485 接口；

- 一个 A/D 接口
- 一个 D/A 接口

图 2-2 系统对外信号接口

2.2.1DSU、RS232 的接口说明

DSU、RS232 分别由两个 DB9 插座引出，接口信号说明如下：

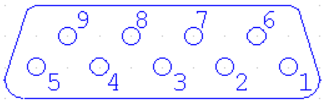


图 2-3 DSU、RS232 插座管脚分布示意图

管脚号	信 号 说 明
2	数据接收
3	数据发送
5	地 线

注：其他未说明管脚悬空。

2.2.2RS422 接口说明

RS422 信号由 J502 凤凰插座引出。

管脚号	信 号 说 明
1	数据发送+
2	数据发送-
3	数据接收+
4	数据接收-

2.2.3RS485 接口说明

RS485 信号由 J501 凤凰插座引出。

管脚号	信 号 说 明
1	信号线正极（A）
2	信号线负极（B）

2.2.4A/D接口说明

A/D 信号由 J1202 凤凰插座引出。

管脚号	信 号 说 明
1	GND
2	模拟量输入

2.2.5D/A接口说明

D/A 信号由 J1203 凤凰插座引出。

管脚号	信 号 说 明
1	GND
2	模拟量输出

2.3 功能模块说明

☞:**对所有功能模块的使用操作可参看光盘提供的对应功能模块的示例程序及相关的技术手册。**

2.3.1S698-MIL处理器芯片

S698-MIL 是高性能、高集成度 SoC 芯片，集成了 32 位 SPARC V8 处理器、浮点处理单元(FPU)、片内 SRAM、SPI 控制器、I^2C 控制器、智能卡控制器等外设，是现代商用设备控制芯片的理想选择。

S698-MIL 主要特征:

- 基于SPARC V8标准的32位RISC高性能嵌入式处理器；
- 4KB指令Cache及4KB数据Cache；
- 内嵌IEEE-754标准的64位浮点处理单元（FPU)；
- 采用AMBA 2.0标准总线作为片内总线；

- 无需仿真器即可支持硬件直接调试（DSU)；
- 片内集成了丰富的外围设备:
  - 存储器控制器；
    - 中断控制器；
    - 片内集成64KB的SRAM；
    - 24位通用定时器和看门狗定时器；
    - IC卡控制器（ISO7816）；
    - I^2C主控器；
    - SPI控制器；
- 最高主频： 180MHz；
- 工作电压： 1.8V（core） & 3.3V（I/O)；
- 产品等级： 商业级、工业级、军品级；
- 芯片封装： QFP160, 塑料或陶瓷封装；

有关 S698-MIL 处理器芯片的详细资料请参阅《S698-MIL 芯片用户手册》。

2.3.2ROM

U301 为 ROM,程序存储器主要用来存储引导程序。512K 字节空间，地址空间从 0x00000000—0x0007FFFF。

2.3.3SRAM

U302、U304 为 SRAM，主要用来运行程序或存储程序运行的临时数据。SRAM 1M 字节空间，2 个 bank，数据总线宽度 8 位（D31 ~ D24)，地址空间从 0x40000000～0x400FFFFF。

2.3.4NAND FLASH

NAND FLASH 为 U303，主要用来存储数据。采用以 SAMSUNG 公司生产的、工业级、大容量、低功耗 3.3V 的 NAND Flash K9F2808U0C（16M x 8bits)，同

时，也兼容更大容量的 K9F5608U0C（32M 字节）和 K9F1208U0A（64M 字节）等 NAND FLASH。

K9F2808U0C 其特性如下：

- K9F2808U0C 的访问时间 50ns，随机存取时间最大值 10us
- 可反复擦写 10 万次,在通常情况下数据可保存 10 年以上

NAND FLASH 读写片选地址为 0x20000005。

S698-MIL 处理器信号定义与 NAND FLASH 控制信号管脚连接如下：

S698-MIL 处理器信号	**NAND FLASH 信号**
GPIO[11]	NAND_ALE
GPIO[12]	NAND_CLE
GPIO[13]	NAND_CS#
GPIO[14]	NAND_R/B#

2.3.5IC卡(又称智能卡)

S698-MIL 内部集成了 3 个智能卡控制器。S698-MIL 智能卡控制器符合 ISO7816 标准。IC 卡输入/输出信号可直接与 CPU 连接。系统中用了 1 个 IC 卡控制器，分别用来读/写用户卡的信息。用户卡是 ID-1 型卡，采用 ID-1 型卡座(U401)。

智能卡接口支持电压有+3.3V 和 5V，系统板通过跳线 J401 来控制智能卡的电压。用跳线帽连接 1 和 2，给智能卡输出电压为+3.3V；用跳线帽连接 2 和 3，给智能卡输出电压为+5V；

智能卡电源开和关，通过 S698-MIL 处理器的 GPIO[27]来控制开和关，GPIO[27]输出‘1’表示开智能卡电源，GPIO[27]输出‘0’表示关智能卡电源；

智能卡(U401)接口信号定义如下：

管脚号	**信号定义**	**管脚号**	**信号定义**
1	VCC	8	NC

2	RST	9	上拉
3	CLK	10	PRES
4	NC	11	NC
5	GND	12	NC
6	NC	13	NC
7	DATA	14	NC

2.3.6DSU串口

S698-MIL 内部集成了软硬件调试单元 DSU(Debug Support Unit)，无需仿真器即可支持硬件直接调试,其外部接口为 DSU 调试串行接口。通过 DSU 口，用户可以访问 CPU 内部所有的寄存器、存储器以及系统所有的存储器和外设等资源，使系统的硬件和软件调试极为方便。

DSU 调试串行接口是一个普通的串行接口（包括收、发和地 3 根信号线），其输入/输出信号（DSURX、DSUTX）均为 TTL 电平，需通过外部电平转换电路将 TTL 电平转换为标准 RS232 电平。

在 DSU 调试模式下，用户可以通过调试环境查看处理器内部的所有资源，包括 IU 内部寄存器、片内寄存器，指令缓存（I-CACHE），数据缓存（D-CACHE）；还可以通过此接口访问开发板上的存储器，实现对存储器的访问、查询、读写操作等；同时，还可以调试应用软件。

2.3.7LCD显示屏

图形点阵 LCD 液晶显示屏 msp-g320240dbcw-21n，分辨率 320 * 240。

LCD 接口(P701)信号定义如下：

管脚号	信号定义	管脚号	信号定义
1	GND	9	D0
2	+5V	10	D1

3	VO	11	D2
4	WR#	12	D3
5	OE#	13	D4
6	CS#	14	D5
7	A0	15	D6
8	RESET#	16	D7

LCD 详细使用和设置，请看 LCD《msp-g320240dbcw-21n》手册。

2.3.8键盘

4*4 的矩阵键盘方便用户使用。采用定时器定时扫描的方式，键盘写寄存器为 0x20000002，键盘读寄存器为 0x20000003。按键分布如下：

0	1	2	↑
3	4	5	↓
6	7	8	Tab
9	.	Del	↵

- ‘0 – 9’数字键：用于数字输入。
- ‘↑’‘↓’键：用来切换界面中的输入项，具有滚动切换的功能。
- ‘.’键：用于小数点输入。
- ‘Tab’键：用于切换设置项中的具体选项。
- ‘Del’键：用于删除输入框中输入的内容。

‘↵’键：用于对当前的操作进行确认。

2.3.9数码管显示

8 位 8 段数码管显示，显示直观，驱动芯片使用 MAX7219。MAX7219 是一种高集成化的串行输入/输出的共阴极 LED 显示驱动器。每片可驱动 8 位 7 段加小数

点的共阴极数码管，可以数片级联，而与微处理器的连接只需 3 根线。MAX7219 内部设有扫描电路，除了更新显示数据时从单片机接收数据外，平时独立工作，极大地节省了 MCU 有限的运行时间和程序资源。

MAX7219 芯片上包括 BCD 译码器、多位扫描电路、段驱动器、位驱动器和用于存放每个数据位的 8×8 静态 RAM 以及数个工作寄存器。通过指令设置这些工作寄存器，可以使 MAX7219 进入不同的工作状态。

采用 SPI 接口。用 S698-MIL 芯片自带的 SPI 接口，编程简单。

S698-MIL 处理器信号定义与 MAX7219 控制信号管脚连接如下：

S698-MIL **处理器信号**	MAX7219 **信号**
GPI1[19]	DIN
GPI1[20]	CLK
GPI1[24]	LOAD

MAX7219 详细资料，请看 MAX7219 器件手册。

2.3.10A/D和D/A

A/D 转换和 D/A 转换是数字世界和模拟世界的桥梁，都使用+2.5V 基准，A/D 芯片使用 MAX1241，输入范围是 0～2.5V；D/A 芯片使用 MAX5302，输出范围是 0～2.5V。A/D 和 D/A 转换芯片都是 SPI 接口，使用 GPIO 口来模拟 SPI 接口。

S698-MIL 处理器信号定义与 MAX1241 控制信号管脚连接如下：

S698-MIL **处理器信号**	MAX1241 **信号**
GPIO[16]	SCLK
GPIO[17]	CS#
GPIO[30]	DOUT

S698-MIL 处理器信号定义与 MAX5302 控制信号管脚连接如下：

S698-MIL **处理器信号**	MAX5302 **信号**

GPI2[0]	CS#
GPI2[1]	DIN
GPI2[2]	SCLK

2.3.11LED

2 个 LED 直接与 GPIO 口相连，3 个 LED 在 IO 空间外扩出，总共 5 个 LED，都是输出电平“0”亮，输出电平“1”暗。

D102	←→	GPIO[0]
D103	←→	GPIO[1]

D201、D202、D203 地址空间在 0x2000000a，这 3 个灯的亮、暗由 0x2000000a 的后 3 位值决定。板上自带状态指示灯(D101),当系统不正常或者没有运行程序是 D101 是亮的，当系统正常或者正在运行程序的话 D101 就不亮。

2.3.12CPLD

CPLD 的主要功能是译码与扩展外部寄存器，外部寄存器列表如下：

地址	Access	寄存器
0x20000002	R/W	key write register
0x20000003	R/W	key read register
0x20000005	R/W	nand flash value register
0x20000006	R/W	lcd value register
0x20000007	R/W	lcd value register
0x2000000a	R/W	led value register
0x2000000b	R/W	lcd cathode voltage control register

第3章 软件系统

3.1 交叉开发调试方式

对于嵌入式系统的开发，其应用软件的开发属于跨平台的开发，所以需要一个集成开发环境。其开发模式是，在一台通用计算机运行软件的编辑编译等操作，然后下载到嵌入式设备中进行运行和调试。用来开发的通用计算机可以选择比较常见的 PC 机、工作站等，运行 Windows 或 Linux 操作系统，开发计算机一般被称为宿主机，而嵌入式设备被称为目标机。在宿主机上编写应用程序，然后通过交叉编译程序编译，下载到目标机上运行，交叉开发环境提供调试工具对目标机上运行的程序进行调试。

S698MIL-DKit 有三种调试方式。其分别是：

- Linux 操作系统下的集成开发环境的交叉调试方式。
- Windows 操作系统下的集成开发环境的交叉调试方式。
- DSU 命令行下的调试方式。

这里主要举例介绍 Windows 操作系统下的集成开发环境的交叉调试方法，然后再介绍 DSU 命令行下的调试方式。请用户在使用前请安装 Orion 4.0 集成开发环境，下面介绍 Orion 4.0 集成开发环境的安装与使用。

3.2 安装 Orion 4.0 集成开发环境

直接双击附带光盘中的 Orion4.0 的安装文件 setup.exe，系统将引导您安装 Orion 4.0，安装过程相对简单，这里不做过多介绍，如图 4-2-1：

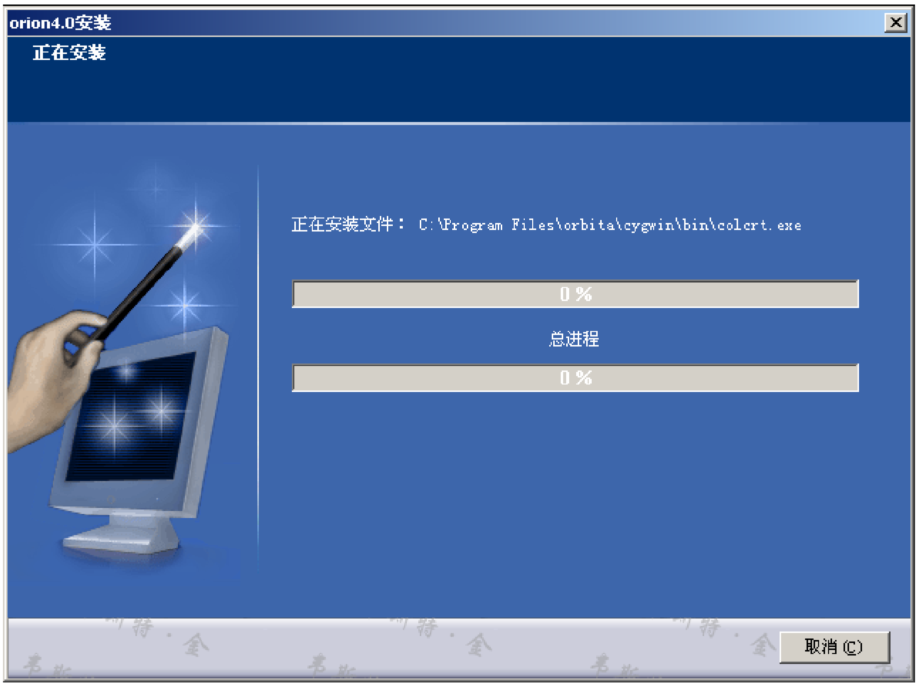


图 4-2-1 应用软件安装界面

3.3 Orion 4.0 使用

在 Orion4.0 软件安装完毕，即可使用该软件，双击 Orion4.0 应用程序图标，如图 4-3-1：

图 4-3-1 应用软件启动界面

3.3.1Orion 4.0 开发流程

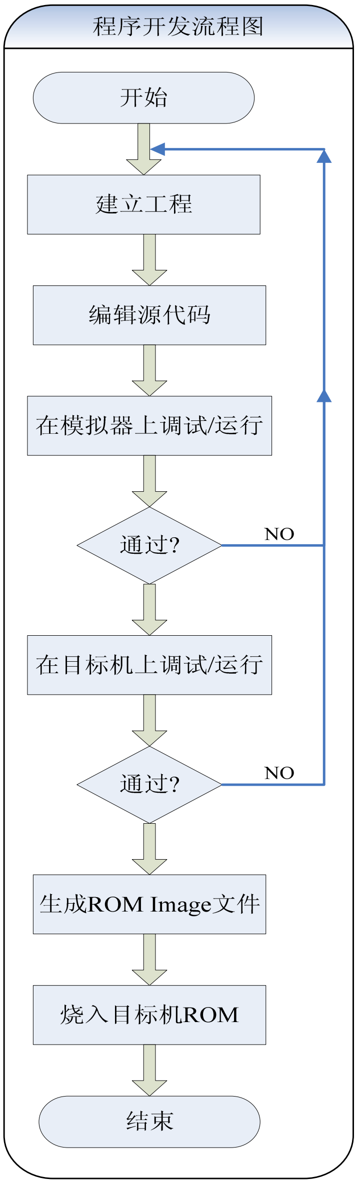
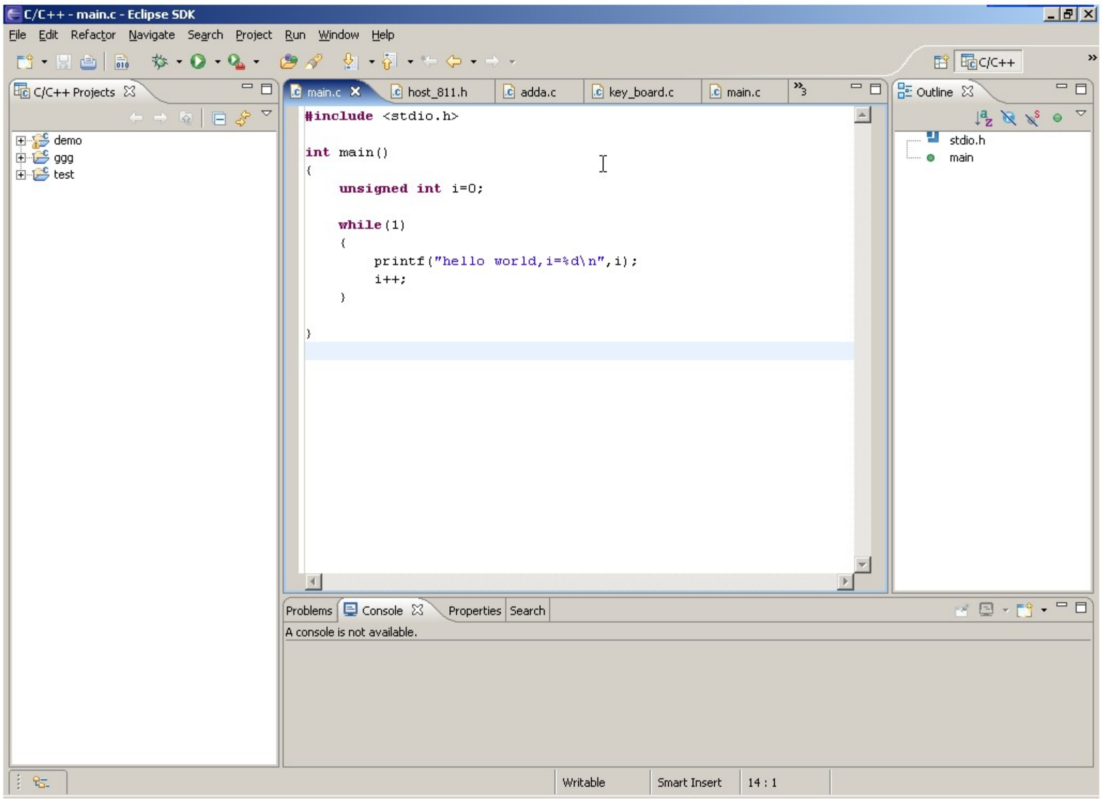


图 4-3-1-1 程序开发流程图

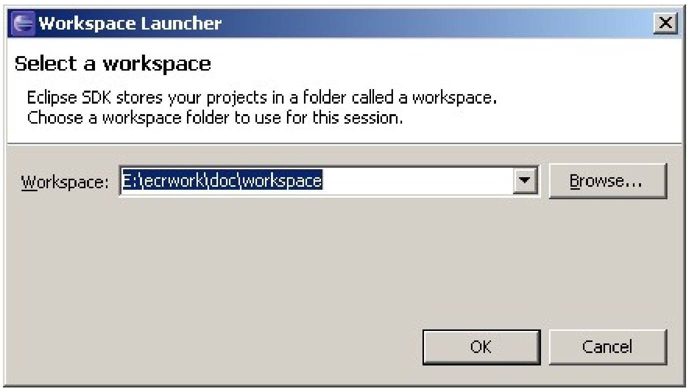
3.4 Orion 4.0 使用说明

3.4.1主窗口



这里先介绍 Orion 4.0 开发环境的主界面，其界面是典型的 windows 风格，与 Visual c++ 等集成开发工具类似；主界面左边显示工程的相关信息和层次关系，右边是代码编辑窗口和编译、调试的返回信息的窗口，一般用户根据其向导很容易创建自己的工程。以下就工程的创建、程序的编译、调试等几个重要部分给出用 Orion 4.0 开发 SPARC 程序的一些简要说明，不详之处请参看 Orion 4.0 的帮助文档。

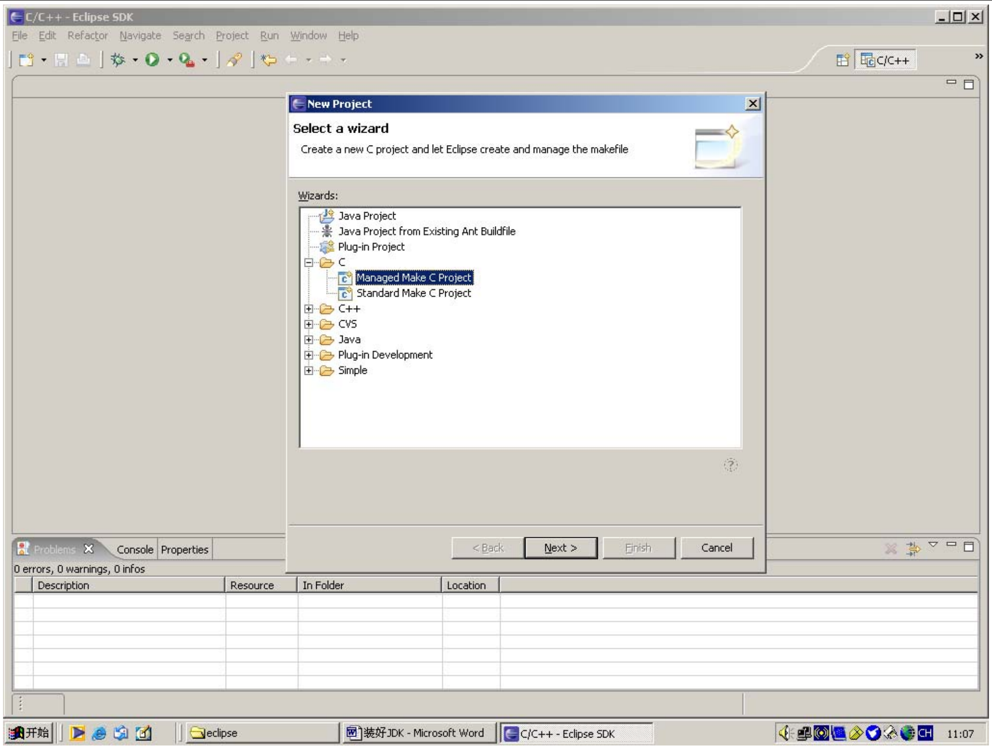
3.4.2选择工作间



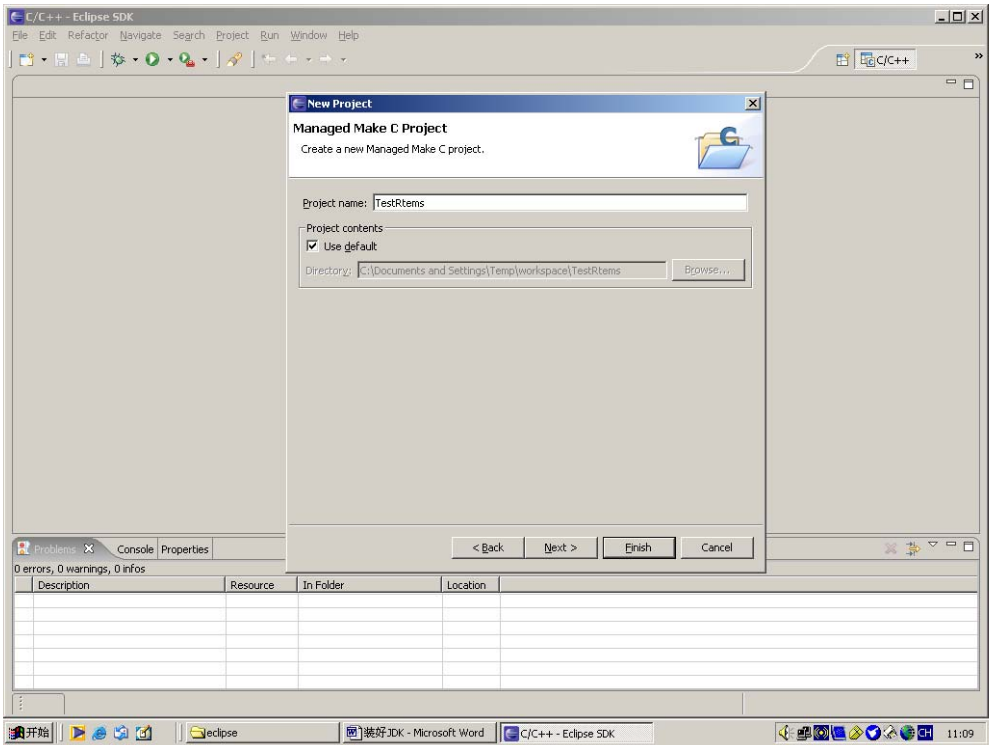
工作间是工程的集合，一个工作间里可以有多个工程；ORION 4.0 登陆时首先要求选择工作间，输入工作间路径后点 OK 进入主界面；此时可以新建工程，打开、关闭已有的工程，但不可以打开其它工作间里的工程；可以点菜单 File->Switch Workspace 切换工作间。**注意：**工作间的路径名中不允许包含空格等特殊字符，否则调试下载时将产生错误。

3.4.3工程的新建

1. 在 ORION 4.0 里按 file/new/project 选择 Manage C Make Project，单击 next 按钮。

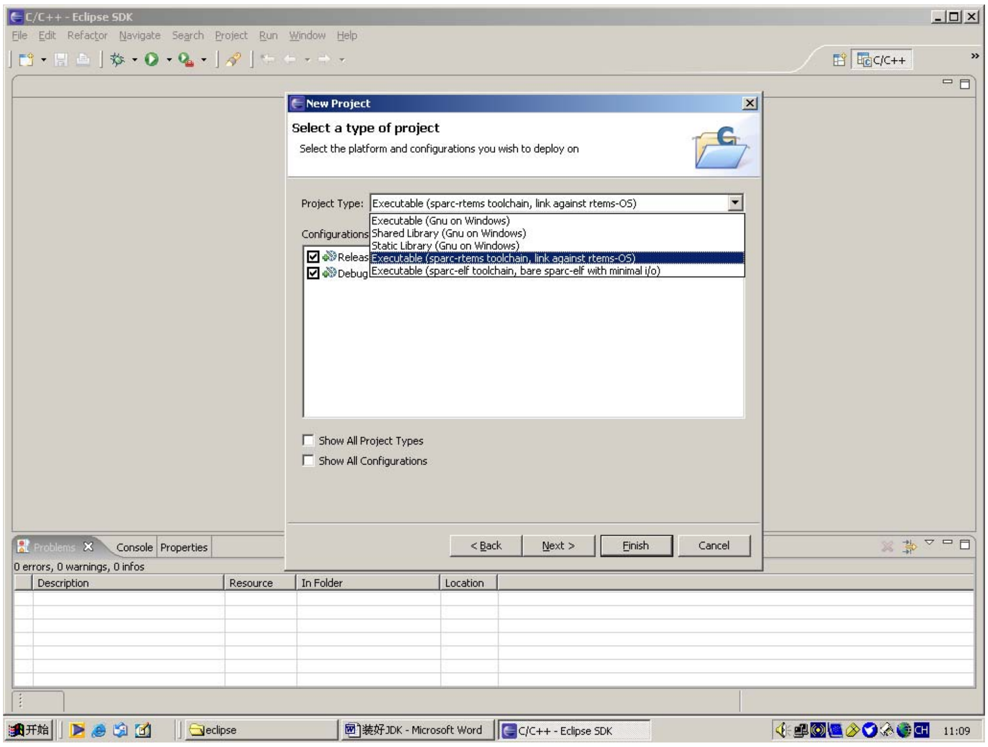


2．在工程名输入框中输入工程名（例图中为 TestRetms），单击 next 按钮。



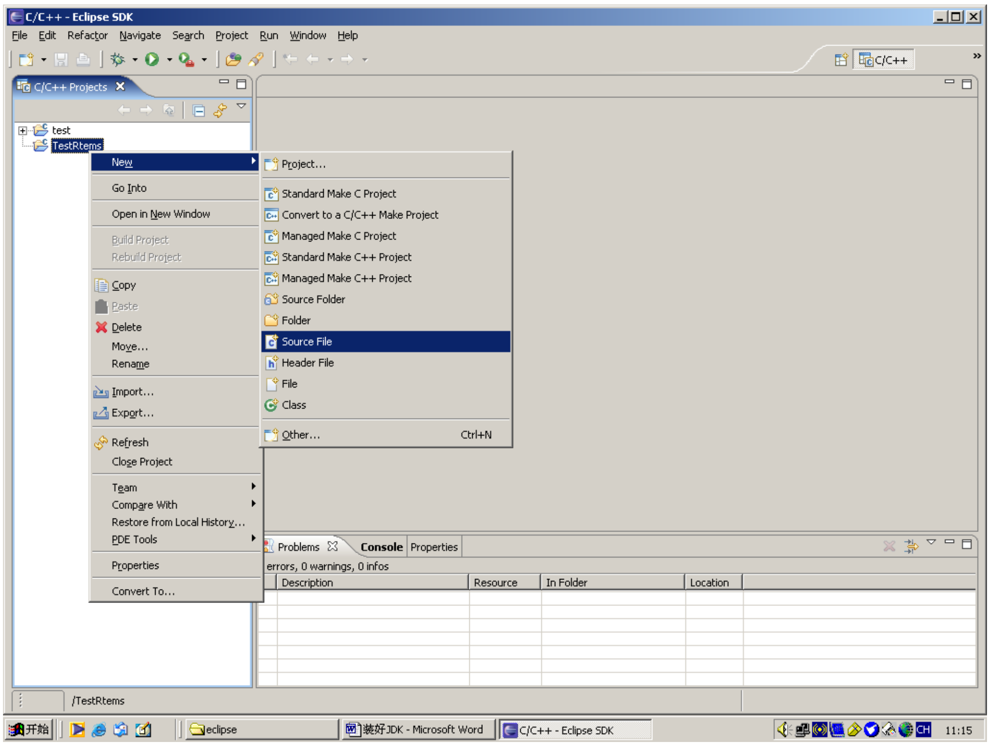
3．在 Project Type 中选择 Excutable(sparc-rtems toolchain,link against

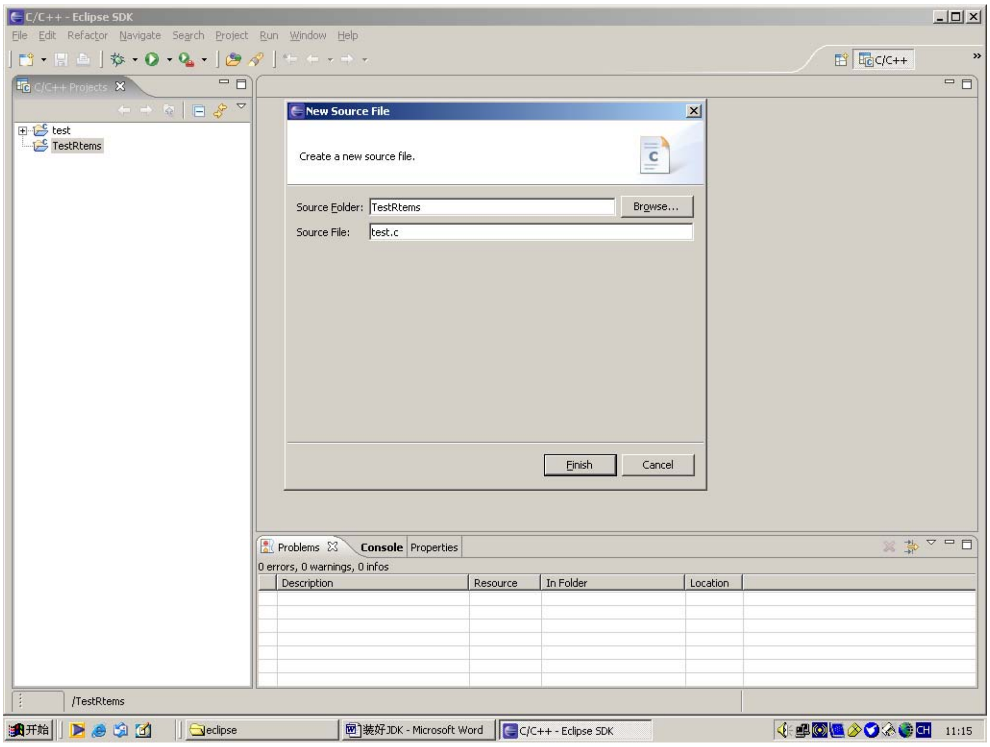
rtems-OS)，单击 next 按钮。



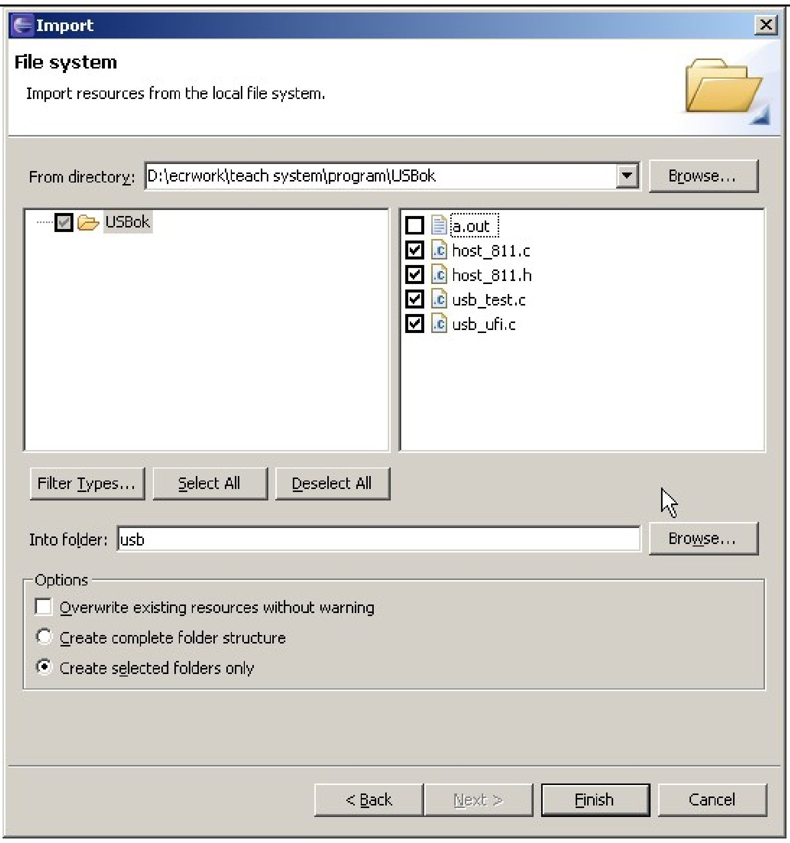
4. 单击 Finish 按钮，建立工程完毕。

5. 新建源文件：在工程导航栏中点右键，选择 New/Source File,输入文件名（例图中为 test.c），单击 Finish 按钮；如果要新建多个源文件，可重复以上步骤依次添加；





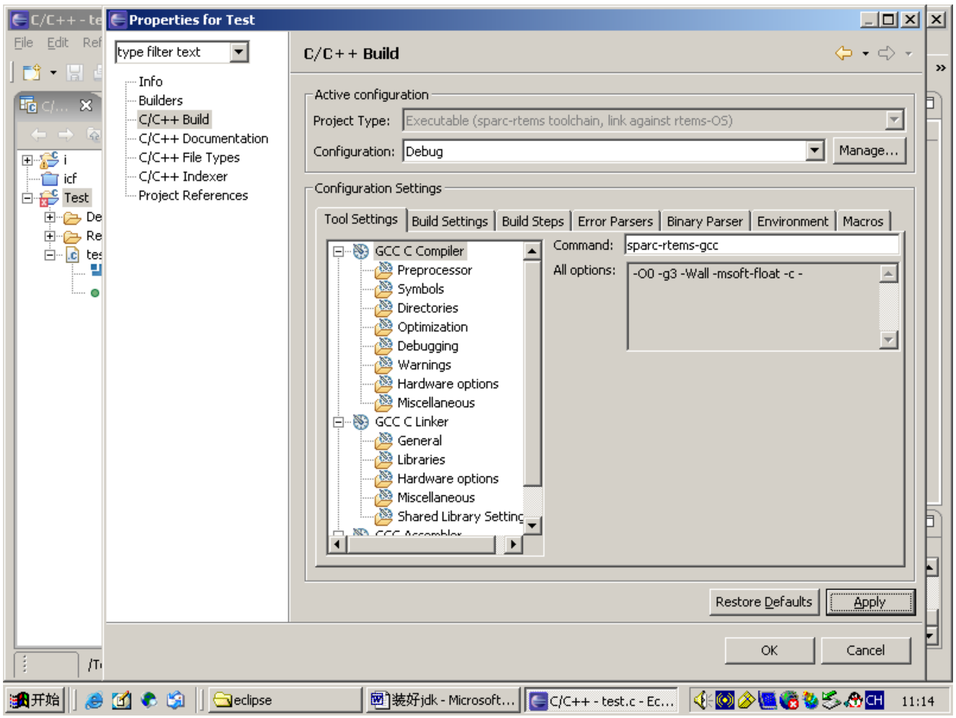
6. 向工程中添加源文件：在工程导航栏选中工程，点右键，浮动菜单中选择 Import 项，弹出文件选择对话框，如下图所示：



单击 Browse 按钮选择文件路径，在右边的文件选择栏中勾选要添加到工程中的文件，单击 Finish 按钮。

3.4.4编译调试环境的配置

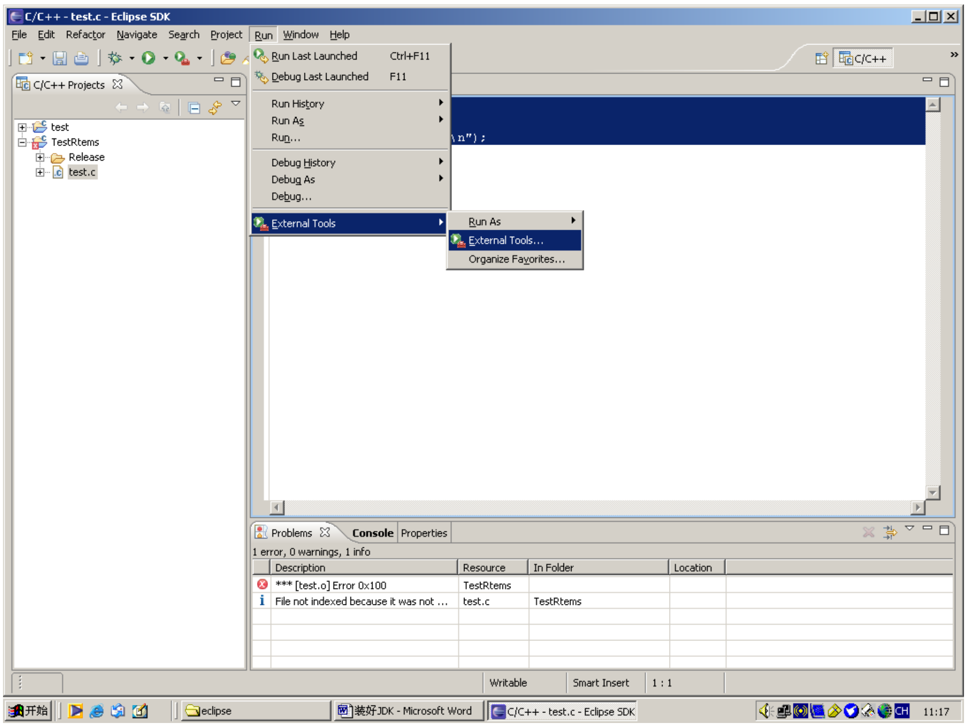
1. 编译环境的配置：选中工程，右键选择 Properties,再选中 C/C++Build，编译选项设置如图：

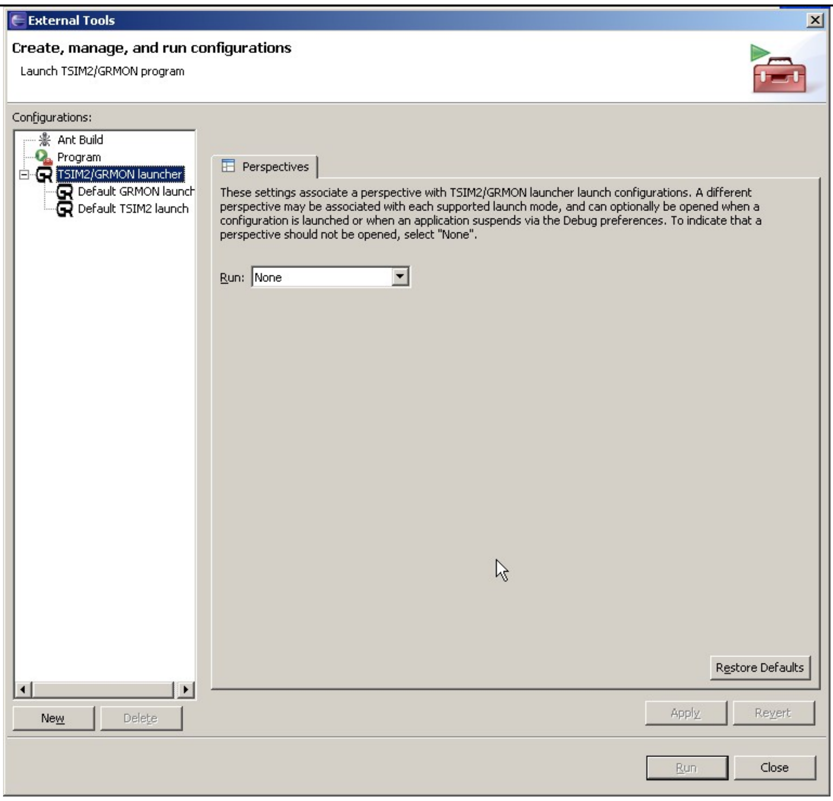


新建的工程系统都给出了默认值，默认值一般不需要改动；如有更改要单击 Apply 按扭，再单击 OK 按扭完成更改；

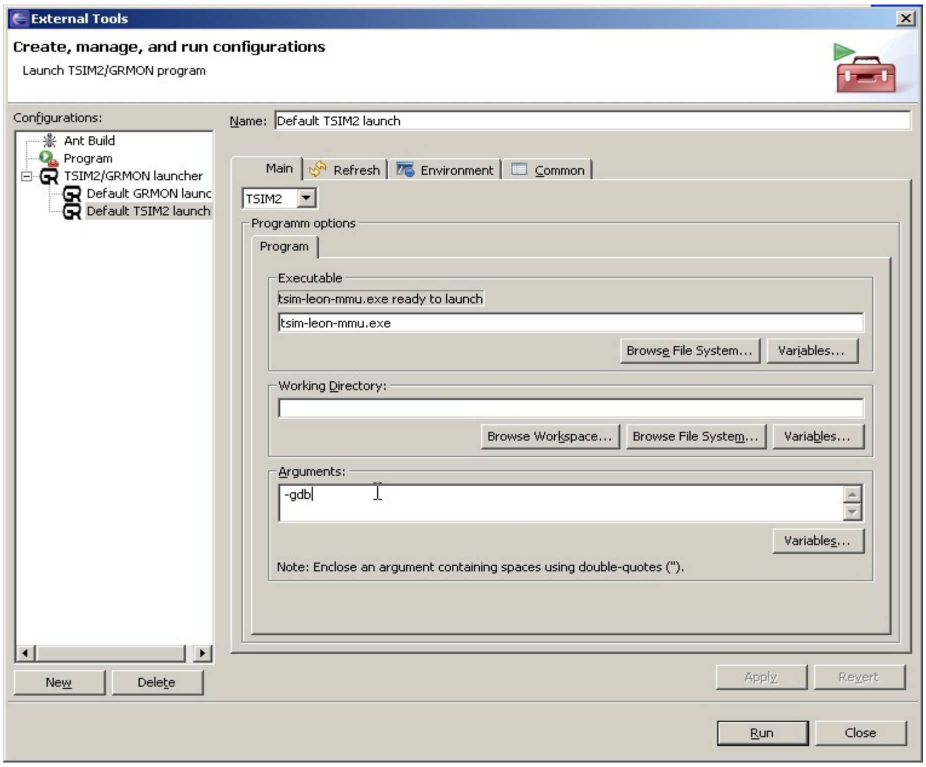
2. 软件模拟器 tsim-ORBITA 的内部启动设置：

(1) 点 run/External Tools/External Tools.





(2) 左边的 Configurations 选择框中选中 Default TSIM2 launch, 在右边的Main页Excutable栏中输入可执行文件名:tsim-ORBITA-mmu.exe,在 Arguments 栏中输入 -gdb ; 其它项保持默认值。



3. 硬件调试器 GRMON 的内部启动设置

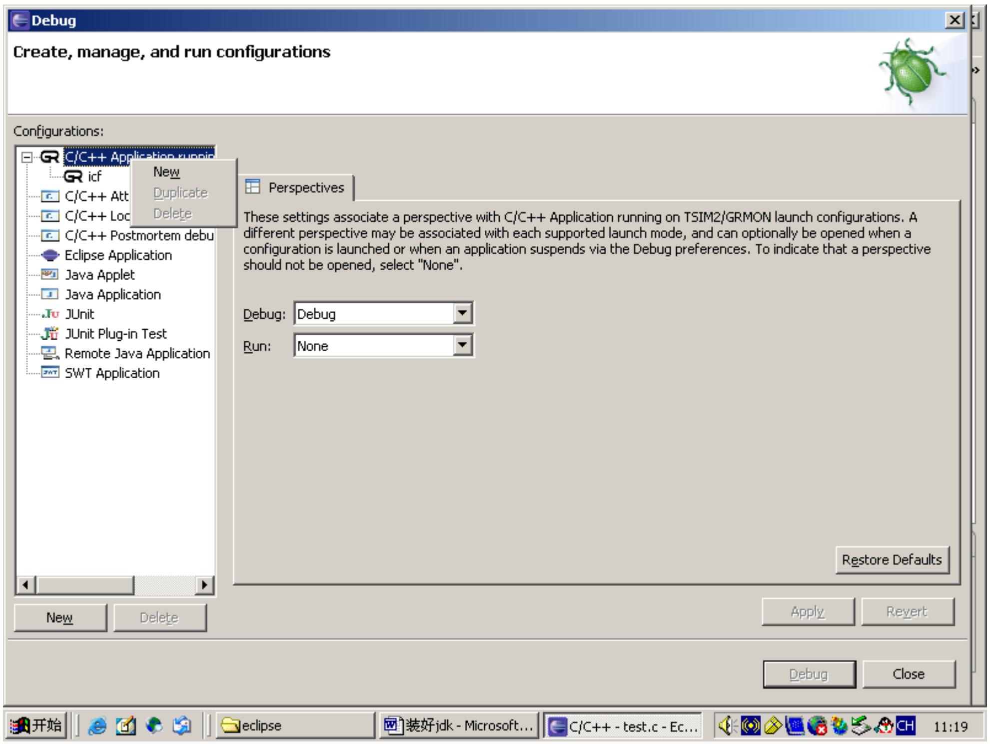
（1） 点 run/External Tools/External Tools;

（2） 左边的 Configurations 选择框中选中 Default GRMON launch, 在右边的 Main 页 Excutable 栏中输入可执行文件名:grmon-eval.exe,在 Arguments 栏中输入 –ORBITA2 –i –u –gdb ; 其它项保持默认值;

（3） 点 Close 按扭结束设置。

3.4.5 调试运行环境的设置

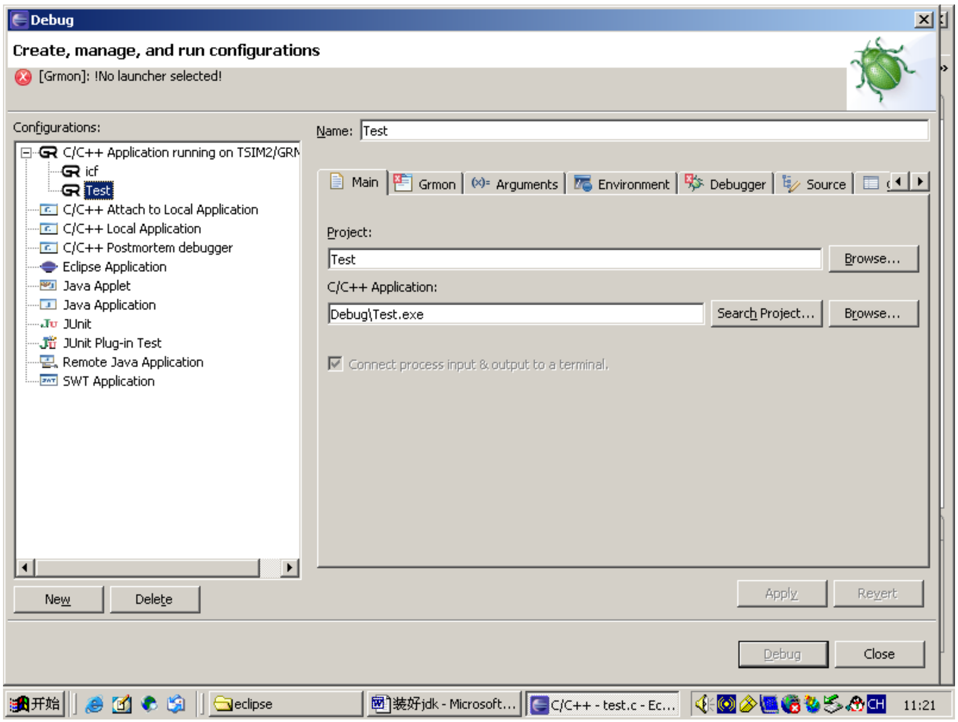
1. 程序调试\运行之前，在工程生成的 EXE 文件中点右键，选择 Debug As，选择 Debug，在弹出视图中的 C/C++ Application running on Tsim/Grmon 项中右键选择 New（或双击）以新建调试\运行环境，如下图：



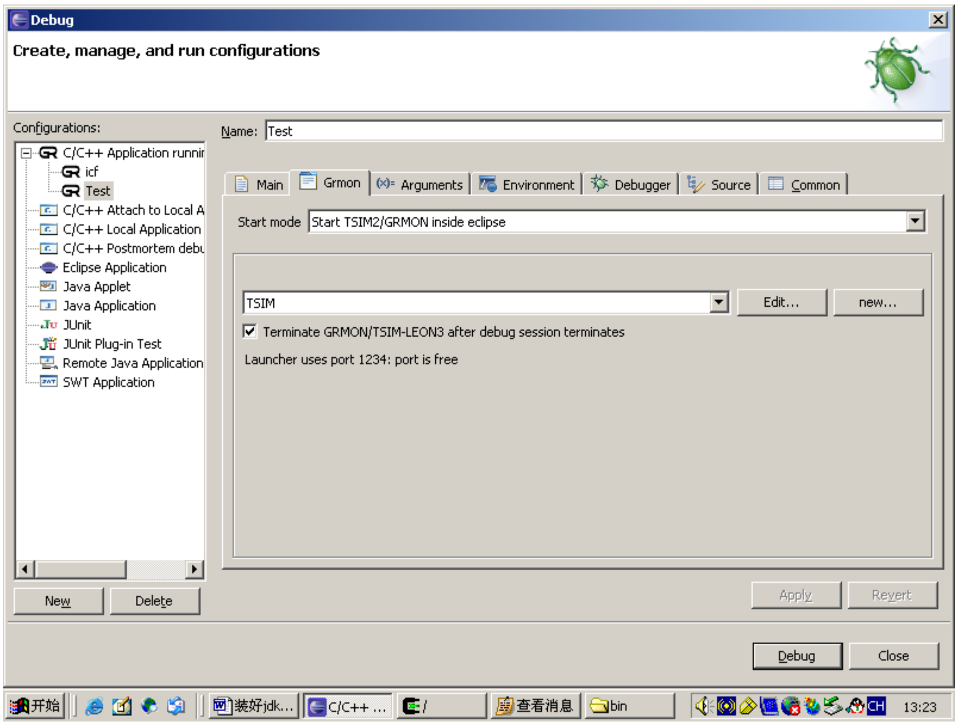
2. Name 输入框中输入新建调试运行环境的名称，Orion 4.0 允许建立多个调试\运行环境，不同的环境之间用用户自定义的名称区分;

3. Main 设置页中指定要调试\运行的工程和可执行文件：Project 输入框中输入

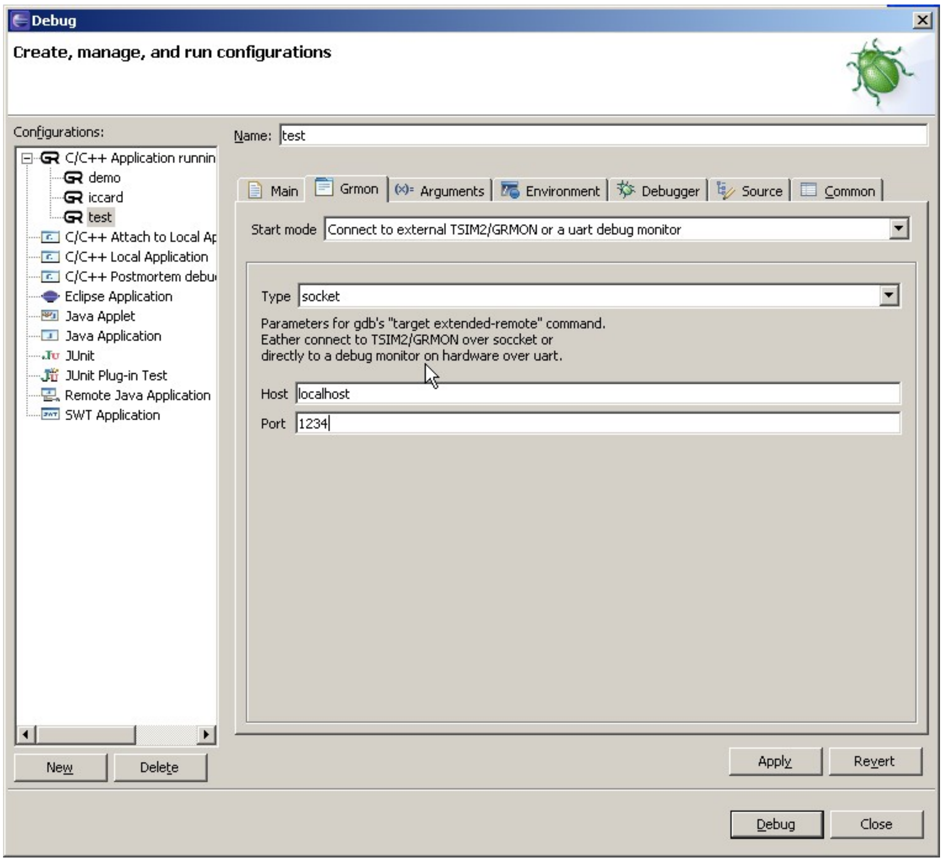
工程名，一般用默认值；C/C++ Application 输入框中输入可执行文件名，可点击后面的 Search Project 按扭选择文件；**注意：**同时有 Debug 和 Release 两个可执行文件时请注意选择，Release 目录下的可执行文件不包含调试信息，运行时不能进行设断点、单步跟踪等操作；



4. Grmon 设置页指定了调试\运行时是用软件模拟器 TSIM 还是用硬件调试器 GRMON，以及调试\运行之前是在 Orion 4.0 内部启动还是在外部启动他们；一般情况下，Start Mode 应选择 Start TSIM2/GRMON inside Orion 4.0,也就是内部启动；然后根据具体情况选择启动 TSIM2 还是 GRMON，选择框中的选择项为上文 run/External Tools/External Tools 中对应的设置名，也可以单击选择框后面的 Edit 和 new 按扭来编辑；



如果选择外部启动 TSIM2/GRMON,则此处要求输入 HOST 和 PORT 值，HOST 中输入值一般为 localhost；如果用 TSIM 调试 PORT 中输入值为 1234，用 GRMON 调试 PORT 中输入值为 2222；外部启动通常用在调试前需要对系统板进行配置的情况。



注意：如果选择外部启动调试器，则在调试之前必须通过 Cygwin 先启动 TSIM 或者 GRMON,命令参数如下：

$ tsim-ORBITA-mmu.exe –gdb

$ grmon-eval.exe -ORBITA2 -i -u -gdb

5. Debugger 设置页中指定了 gdb 调试器的一些参数，具体设置如下：

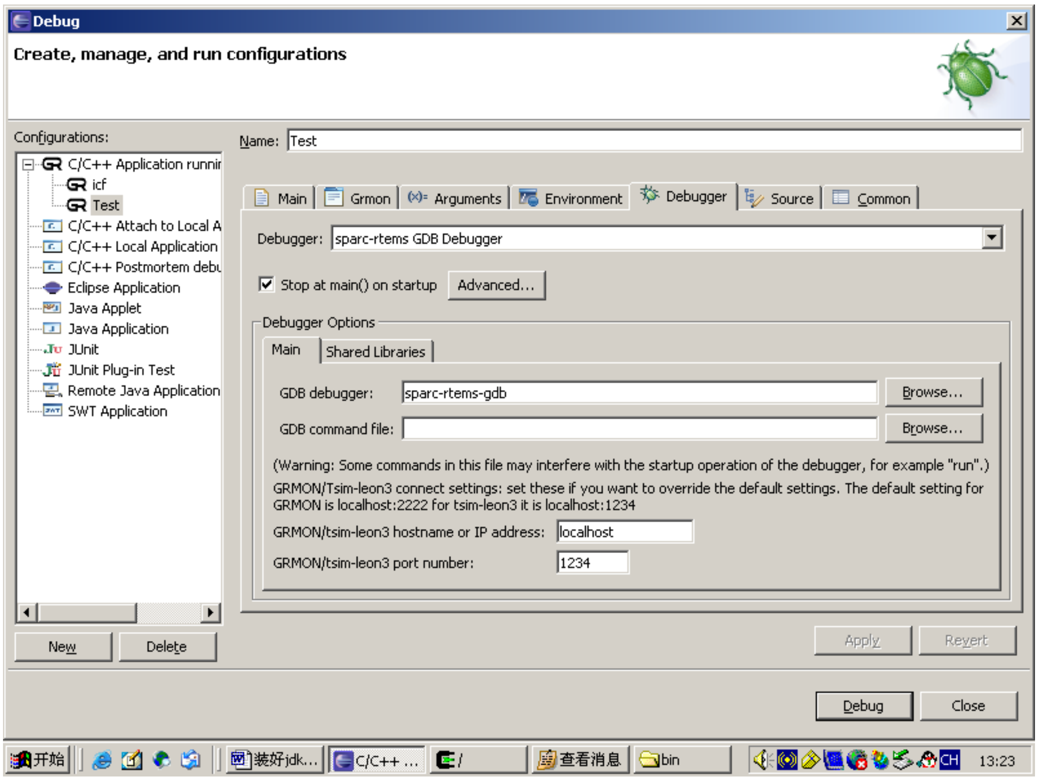
(1) Debugger 选择框中选 sparc-rtems GDB Debugger；

(2) Main 子页中 GDB Debugger 输入框中输入 sparc-rtems-gdb；

(3) GRMON/tsim-ORBITA3 hostname or IP address 中输入 localhost ，既本地调试；

(4) GRMON/tsim-ORBITA3 port name 中输入 gdb 通讯时对应的 Socket 端口号，软件模拟器 TSIM 的端口号为 1234，硬件调试器 GRMON 的端口号为 2222；

(5) 其它文中未提到的设置项请用默认值。



3.4.6工程的编译、链接

编写完程序后就可以开始编译了，编译成功的前提是编译环境的正确配置。

默认情况下的 Orion 4.0 是“自动编译”，就是一保存原文件就会自动编译，不喜欢的可以取消 project 菜单下的 Build Automatically 项，待需要编译时点击工具栏中的 Build All 按扭 来编译工程。

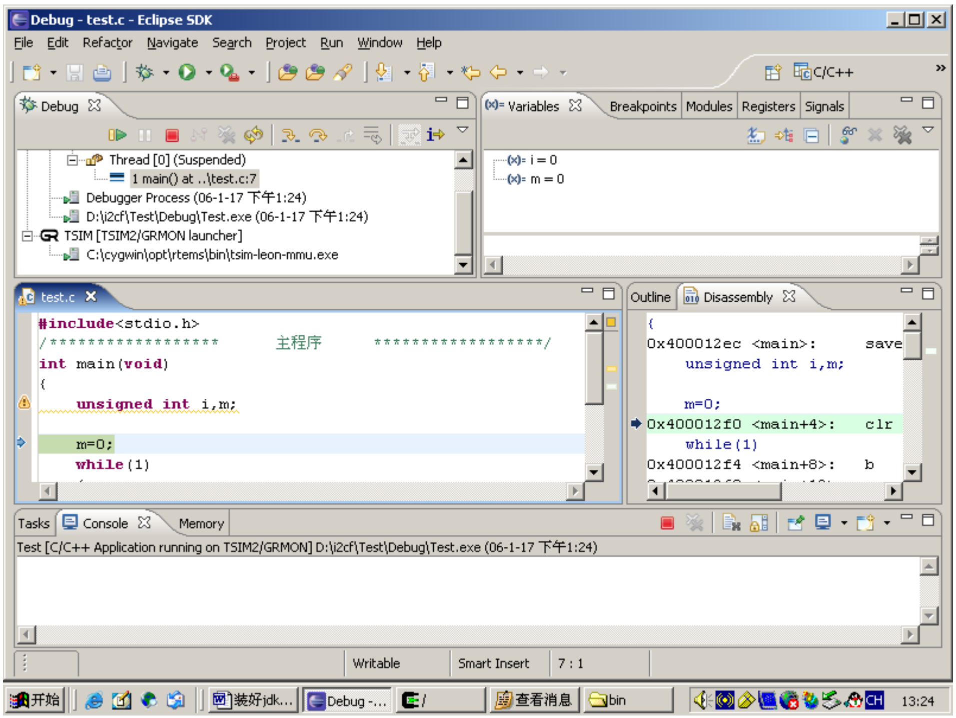
3.4.7程序的调试

程序编译成功后就剩下的工作就是调试，Orion 4.0 具有强大的调试功能，调试步骤和主要辅助功能简单介绍如下：

1. 启动调试

调试环境设置好后单击 Debug 按扭，启动调试（如果设置了多个调试环境，单击 Debug 按扭默认启用第一个调试环境），此时主界面切换到 Debug 视图（如下图所示，如果没有进入到 Debug 视图得话可在菜单 Window/Open

Perspective/Other/Debug 打开 Debug 视图)，主界面下方的 Console 输出信息框中提示调试连接的状况，如果连接失败请检查设置选项；注意：硬件调试前请正确连接硬件系统。



2. 设置断点

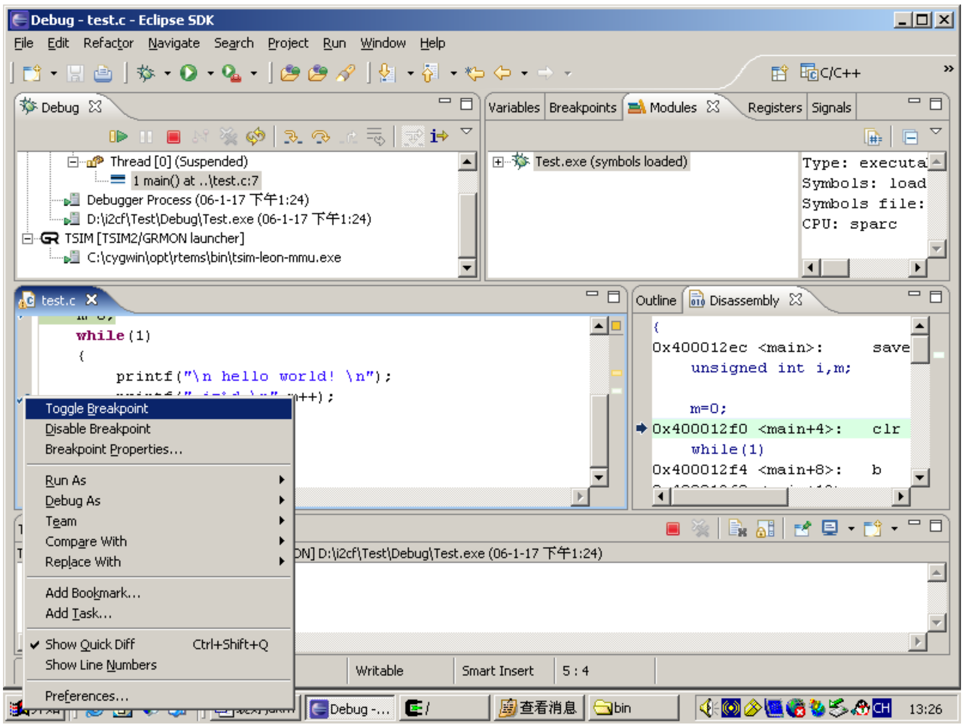
设置断点通常有以下几种方法：

(1) 将鼠标指向源程序行左边需设定断点处，点右键，在浮动菜单中选择 Toggle Breakpoint;

(2) 将光标移动到需要设置断点的代码行处，选择 Run/Toggle Line Breakpoint;

(3) 将光标移动到需要设置断点的代码行处，按快截键 Ctrl+shift+B;

删除断点的方法如此类似，当程序运行到断点处时，会停止在有效断点处，源程序左边出现指示箭头；建议调试开始之前先设置好断点，可防止一启动调试后程序就立刻运行完毕。



3. 单步运行

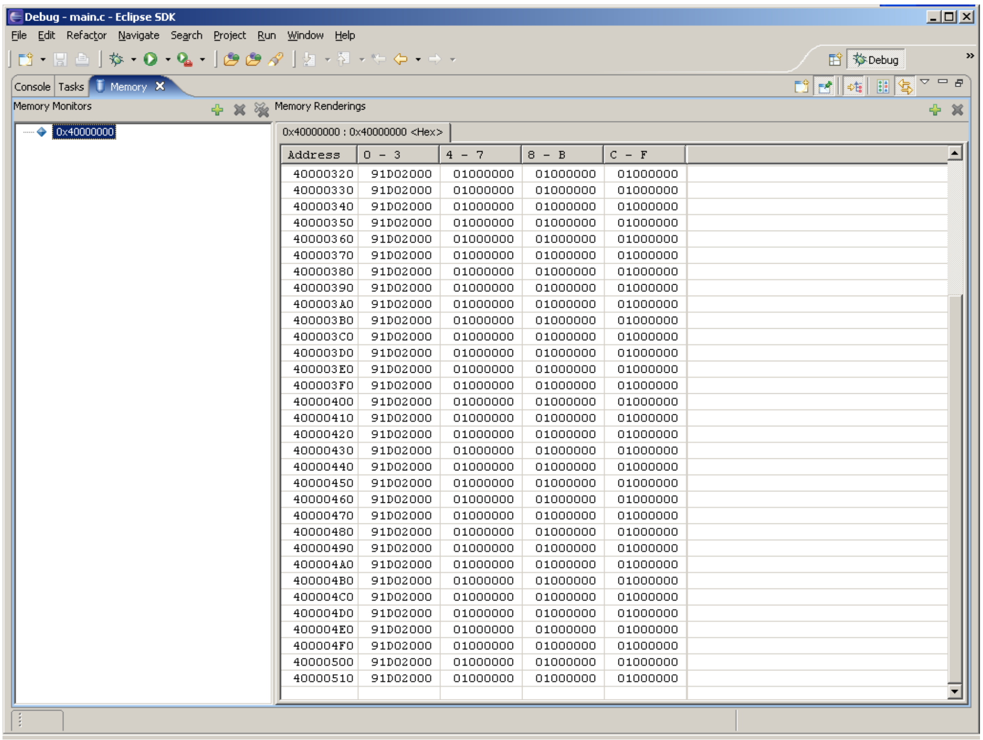
程序运行到断点处后，可单击 Step Into 按扭 或 Step Over 按扭 进行单步运行，其中 Step Into 是运行到函数处会进入函数内部，Step Over 则会跳过函数运行函数后的程序行；源程序左边的箭头指示当前正待运行的程序行。

4. 观察变量

选择 Windows/Show View/Variables 项，可打开变量观察窗，在其中点击右键可进行添加删除观察变量的操作，Format 项用来选择数据显示的方式（10 进制、16 进制）；如果所观察的变量值改变，则系统以红色显示该变量。

5. 观察内存

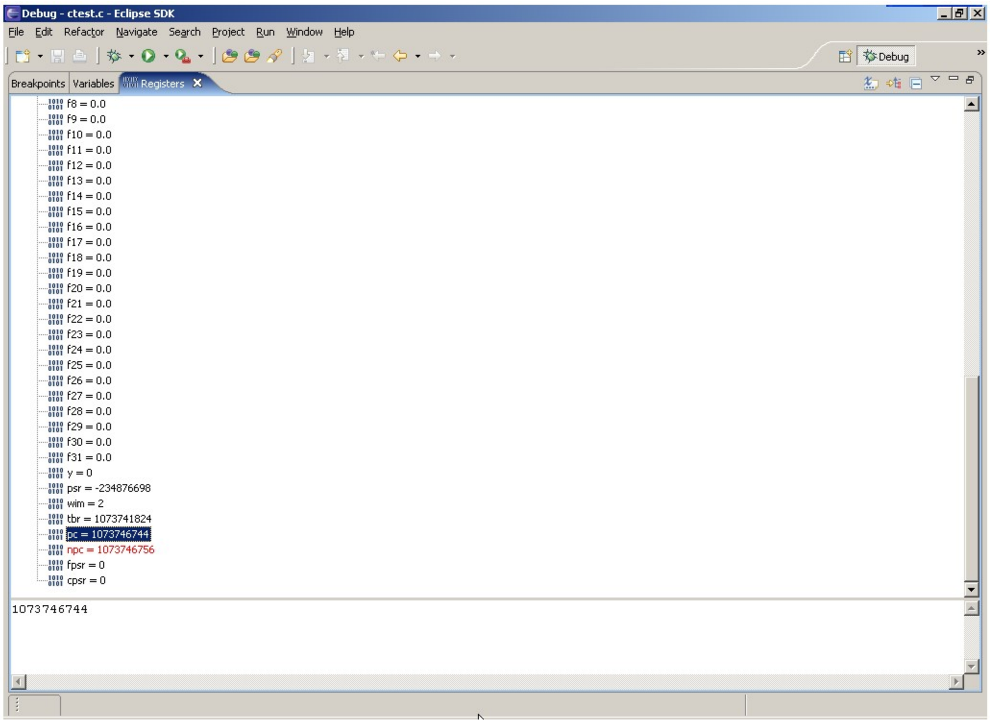
选择 Windows/Show View/Memory 项，可打开内存观察窗，单击 Add Memory Monitor 按扭输入要观察的内存起始地址（建议输入十六进制如：0x40000000），点 OK 按扭后，右边界面将显示当前内存地址的值，数据以 16 个字节为一行依次显示，拖动滚动条系统将按照对应地址提取数据显示，内存显示如图：



Address	0 - 3	4 - 7	8 - B	C - F
40000320	91D02000	01000000	01000000	01000000
40000330	91D02000	01000000	01000000	01000000
40000340	91D02000	01000000	01000000	01000000
40000350	91D02000	01000000	01000000	01000000
40000360	91D02000	01000000	01000000	01000000
40000370	91D02000	01000000	01000000	01000000
40000380	91D02000	01000000	01000000	01000000
40000390	91D02000	01000000	01000000	01000000
400003A0	91D02000	01000000	01000000	01000000
400003B0	91D02000	01000000	01000000	01000000
400003C0	91D02000	01000000	01000000	01000000
400003D0	91D02000	01000000	01000000	01000000
400003E0	91D02000	01000000	01000000	01000000
400003F0	91D02000	01000000	01000000	01000000
40000400	91D02000	01000000	01000000	01000000
40000410	91D02000	01000000	01000000	01000000
40000420	91D02000	01000000	01000000	01000000
40000430	91D02000	01000000	01000000	01000000
40000440	91D02000	01000000	01000000	01000000
40000450	91D02000	01000000	01000000	01000000
40000460	91D02000	01000000	01000000	01000000
40000470	91D02000	01000000	01000000	01000000
40000480	91D02000	01000000	01000000	01000000
40000490	91D02000	01000000	01000000	01000000
400004A0	91D02000	01000000	01000000	01000000
400004B0	91D02000	01000000	01000000	01000000
400004C0	91D02000	01000000	01000000	01000000
400004D0	91D02000	01000000	01000000	01000000
400004E0	91D02000	01000000	01000000	01000000
400004F0	91D02000	01000000	01000000	01000000
40000500	91D02000	01000000	01000000	01000000
40000510	91D02000	01000000	01000000	01000000

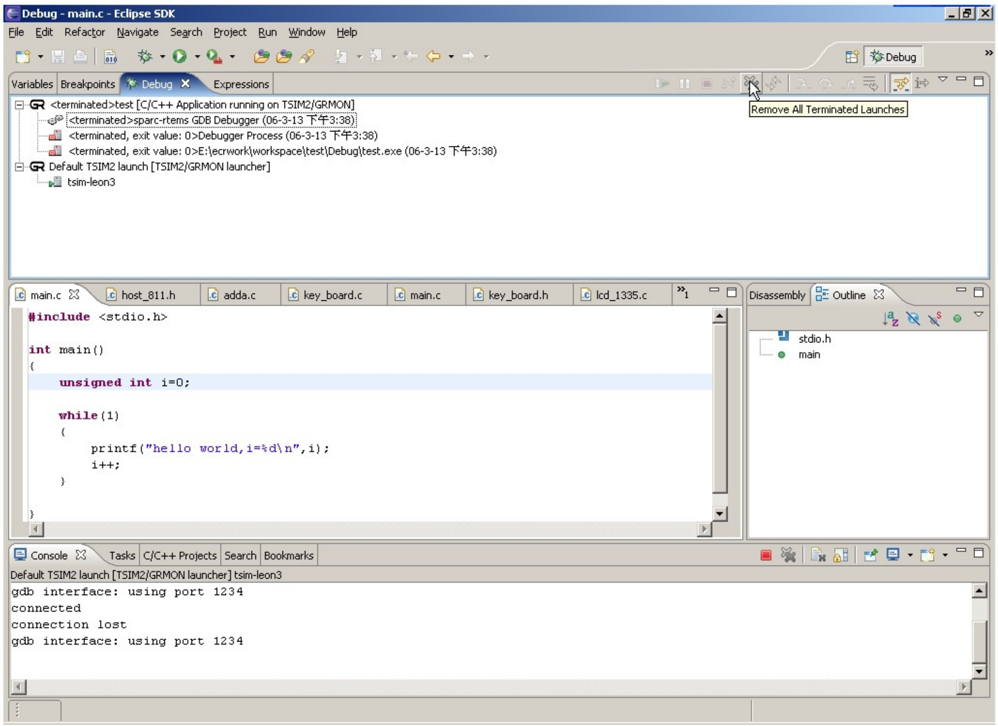
6. 观察寄存器

选择 Windows/Show View/Registers 项,可打开寄存器观察窗，如下图；随着程序运行到不同的程序行，可以观察到各寄存器所产生的变化；如果寄存器值改变，则系统以红色显示该寄存器。



7. 删除调试

调试完成或者中途需要重新开始调试时要将当前的调试进程关闭；单击Debug窗口或者Console窗口右上方的Remove All Terminated Launches按扭 可完成删除操作，如果调试进程仍在运行需要先单击 terminate 按扭 终止调试后再进行删除。



3.5 DSU调试步骤

S698-MIL 芯片内部集成了专门用来调试系统软、硬件的 DSU（Debug Support Unit）调试接口。DSU 是 SPARC V8 架构所特有的硬件调试接口，通过 DSU，用户可以对 CPU 内部所有的寄存器和系统所有的存储设备进行读、写操作，并且可以设置断点，从而使系统软、硬件的调试变的极为方便。

DSU 的详细资料请参阅《DSU Monitor 用户手册》使用手册。

使 CPU 进入 DSU 调试模式的步骤如下所示：

1. 关闭电源。
2. 将系统板上 P502 DSU 调试串口通过串口转接线连接到主机的 COM1 口。
3. 加电启动

6. 在 cygwin 的窗口中，输入 dsumon －i －u 命令后，主机通过串口 1 与系统板的 DSU 调试串口（dsurx，dsutx）建立通信连接（此时，用户可通过主机对目标板的 CPU 进行调试）。dsumon 命令连接成功后会提示如下的信息：

```
$ dsumon.exe -i -u

 ORBITA DSU Monitor 1.0.8  (evaluation version)
```

```
 Copyright (C) 2003, ORBITA - all rights reserved.
 Comments or bug-reports to jiri@gaisler.com

 using port /dev/ttyS0 @ 115200 baud

 processor frequency   :  20.28 MHz
 register windows      :  8
 v8 hardware mul/div   :  yes
 floating-point unit   :  meiko
 instruction cache     :  1 * 4 kbytes, 32 bytes/line (4 kbytes total)
 data cache            :  1 * 4 kbytes, 32 bytes/line (4 kbytes total)
 hardware breakpoints  :  2
 sram width            :  8 bits
 sram banks            :  2
 sram bank size        :  512 kbytes
 stack pointer         :  0x400ffff0
 UART 1 in DSU mode
dsu> lo flash.out
section: .text at 0x40000000, size 66400 bytes
section: .data at 0x40010360, size 1904 bytes
total size: 68304 bytes (90.5 kbit/s)
dsu>
```

7. 在 DSU>提示符后，使用 lo 命令将编译好的可执行文件（*.out）下载到目标板的 SRAM 区。信息如下：

```
dsu> lo test.out
section: .text at 0x40000000, size 66400 bytes
section: .data at 0x40010360, size 1904 bytes
total size: 68304 bytes (90.5 kbit/s)
dsu>
```

7. 在 DSU>提示符后，使用 run 命令使目标板 SRAM 中的程序开始运行。在运行过程中，用户可通过 ctrl+c 使程序停止运行。

3.5.1FLASH程序固化

采用 FLASH 来作为引导程序，需要把引导程序、操作系统和应用程序下载到 FLASH。

下面以应用程序 lcdc.c 为例，说明如何将程序烧录到 FLASH 中。在 cygwin 的窗口中。

1. 编译文件和烧写程序，步骤如下：

- 编译文件

```
$ sparc-rtems-gcc lcdc.c
$ mkprom.exe -freq 20 -baud 38400 -ramsize 1024 -ramcs 2 -ramws 15 -ramwidth 8
-romsize 512 -romwidth 8 a.out -o a.prom

MKPROM boot-prom builder v1.3.8
Copyright ORBITA 2001, all rights reserved.

section: .text at 0x40000000, size 33920 bytes
section: .data at 0x40008480, size 69104 bytes

creating ORBITA boot prom: a.prom

$ sparc-rtems-objcopy -O binary  a.prom a.bin

Administrator@sky ~/JXXT/program/LCDok
$ mkflash -sst -width 8 a.bin

MKFLASH - S698 flash-download builder v2.0.0, copyright Orbita inc, 2004.08.03
Bug-reports to yc_gyh@orbitabluebox.com

loading: a.bin.
file stream length:33072 byte.

creating S698 flash download: flash.out
```

- 烧写程序

```
$ dsumon.exe -i -u

 ORBITA DSU Monitor 1.0.8  (evaluation version)

 Copyright (C) 2003, ORBITA - all rights reserved.
 Comments or bug-reports to jiri@gaisler.com

 using port /dev/ttyS0 @ 115200 baud

 processor frequency   :  20.28 MHz
 register windows      :  8
 v8 hardware mul/div   :  yes
 floating-point unit   :  meiko
 instruction cache     :  1 * 4 kbytes, 32 bytes/line (4 kbytes total)
 data cache            :  1 * 4 kbytes, 32 bytes/line (4 kbytes total)
 hardware breakpoints  :  2
 sram width            :  8 bits
 sram banks            :  2
 sram bank size        :  512 kbytes
 stack pointer         :  0x400ffff0
 UART 1 in DSU mode
```

```
dsu> lo flash.out
section: .text at 0x40000000, size 66400 bytes
section: .data at 0x40010360, size 1904 bytes
total size: 68304 bytes (90.5 kbit/s)
dsu> ru
erasing the flash,Please wait.
            erase finished.

            begin write to flash,wait.
            .....................
            end write to flash.
            write total 20656 bytes data to flash.
            Program exited normally.
dsu>
```

执行以上步骤后，程序已固化到 FLASH 中，关闭电源，重新上电，则程序将自动运行。光盘中带有系统默认烧写文件，如果用户不小心修改了 FLASH 信息，可重新烧写。

3.6 菜单界面

系统自带有默认软件，通电就可使用。下面分别介绍 S698MIL-DKit 液晶屏各菜单界面。

3.6.1LED界面

LED 灯控制：LED 灯按扭为乒乓开关，完成灯的打开和关闭操作；先敲 Tab 键切换到 1、2、3、4 号灯按扭，再敲回车键，板上对应的 LED 将点亮或者熄灭。

LED 数码管控制：可以输入 0123456789 不同的数字让数码管显示不同的数字；

输入数值（最多 8 位，小数点占一位）后，选中“显示开”按扭，敲回车键。例如：

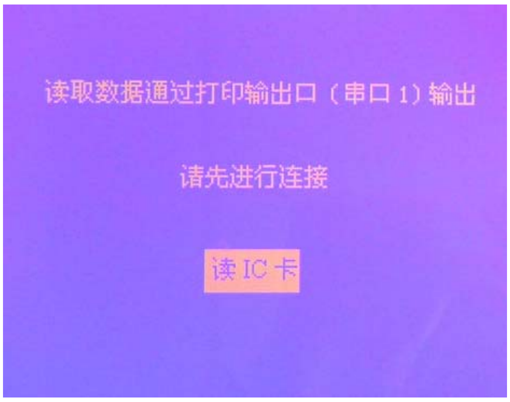
3.6.2A/D D/A界面

A/D 输入：当 A/D 端口有电压输入时，相应的“当前 A/D 输入电压：[ ]伏”就会显示相应的电压；

D/A 输出：如果你想输出几伏的电压，在“D/A 输出电压：[ ]”键入要输出的电压值，敲回车键就可以了，此时可以用电压表在 D/A 端口量取 D/A 输出值是否键入的值相符。

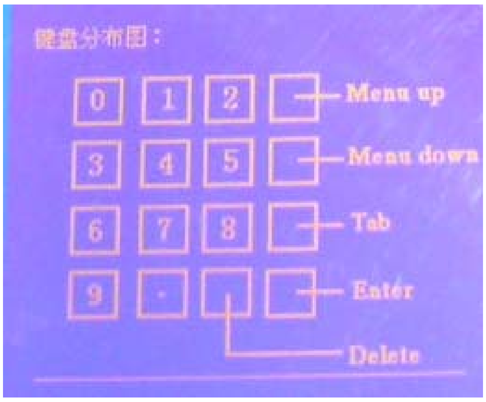
注意：A/D 输入范围是 0～2.5V，输入电压请不要超过那范围。

3.6.3IC卡界面



- 请用串口线把系统板的串口 1 和 PC 机串口连接起来，打开串口调试助手，把波特率调到 38400，插入 IC 卡，敲回车键，系统板就会把 IC 卡的信息发到 PC 机。系统板上默认的时 5V 的 IC 卡接口（系统板自带 IC 卡为 5V）。如果使用 3.3V 的 IC 卡，请把跳线帽接在 J401 的 PIN 1-2。

3.6.4帮助界面



第4章 维护和故障处理

4.1 日常维护

为了使系统长期良好运行，在日常维护中应做到以下几点：

- 应尽可能使用在条件良好的环境，周围没有强烈震动、热辐射、强烈光照、灰尘较多以及高温度、高湿度的场合；
- 定期检查系统的运行情况；
- 保持系统的卫生，避免上面堆积灰尘。

4.2 故障处理

下面列出了一些可能出现的故障现象以及处理方法，如果系统出现了其它故障，请速与厂家联系。

故障现象	处理方法
开机无显示，电源指示灯灭	检查输入 220V AC 是否在设备允许的范围内，如果输入不正常，应立即关闭输入电源，以免损坏设备。如果输入正常，设备有故障，请和厂家联系。
按键操作无效	关闭设备电源，重新开机，如还不能正常工作，设备有故障，请和厂家联系。

附录 A：订货信息

订货信息表

型号	说明
S698MIL-DKit	S698-MIL 应用开发系统

附录 B：装箱单

附表一： 装 箱 单

序号	物品名称	单位	数量	备注
1	S698MIL-DKit 系统	台	1	
2	串口线	根	2	
3	IC 卡	个	1	
4	资料光盘	张	1	
5	合格证	张	1	